From Coarse to Fine: Hierarchical Pixel Integration for Lightweight Image Super-resolution

Jie Liu*, Chao Chen†, Jie Tang†, Gangshan Wu
State Key Laboratory for Novel Software Technology, Nanjing University, China
liujie@nju.edu.cn, chenchao@smail.nju.edu.cn, {tangjie,gswu}@nju.edu.cn

Abstract
Image super-resolution (SR) serves as a fundamental tool for the processing and transmission of multimedia data. Recently, Transformer-based models have achieved competitive performances in image SR. They divide images into fixed-size patches and apply self-attention on these patches to model long-range dependencies among pixels. However, this architecture design is originated for high-level vision tasks, which lacks design guideline from SR knowledge. In this paper, we aim to design a new attention block whose insights are from the interpretation of Local Attribution Map (LAM) for SR networks. Specifically, LAM presents a hierarchical importance map where the most important pixels are located in a fine area of a patch and some less important pixels are spread in a coarse area of the whole image. To access pixels in the coarse area, instead of using a very large patch size, we propose a lightweight Global Pixel Access (GPA) module that applies cross-attention with the most similar patch in an image. In the fine area, we use an Intra-Patch Self-Attention (IPSA) module to model long-range pixel dependencies in a local patch, and then a spatial convolution is applied to process the finest details. In addition, a Cascaded Patch Division (CPD) strategy is proposed to enhance perceptual quality of recovered images. Extensive experiments suggest that our method outperforms state-of-the-art lightweight SR methods by a large margin. Code is available at https://github.com/passerer/HPINet.

Introduction
Single-Image Super-Resolution (SISR) aims to recover a visually pleasing high-Resolution (HR) image from its Low-Resolution (LR) counterpart. SISR is widely used in many multimedia applications such as facial recognition on low-resolution images and server costs reduction for media transmission. With the success of Convolutional Neural Networks (CNNs), CNN-based models (Zhang, Zeng, and Zhang 2021; Dong, Loy, and Tang 2016; Tai et al. 2017; Zhang, Zuo, and Zhang 2018; Li et al. 2019; Lim et al. 2017; Zhang et al. 2018c) have become the mainstream of SISR due to the natural local processing ability of convolution kernels.

1These authors contributed equally.
†Corresponding author.
Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.
In summary, our main contributions are as follows:
• We introduce a hierarchical interpretation of the Local Attention Map (LAM) (Gu and Dong 2021) and devise a new attention block for image SR.
• Instead of the vanilla patch division method that fixes the patch size throughout the network, a Cascaded Patch Division (CPD) strategy is applied for better perceptual quality in terms of LPIPS.
• We propose a lightweight Hierarchical Pixel Access Network (HPINet) that outperforms existing lightweight SR methods by a large margin. Most notably, our HPINet achieves results that match state-of-the-art large (around 15M) SR models using less than 1.5M parameters.

Related Work

Deep Neural Networks for SR

Deep neural networks have become the most popular methodology for image SR in the past several years thanks to their great representation power. Since (Dong et al. 2014) first uses three convolution layers to map the low-resolution images to high-resolution images, various CNN-based networks have further enhanced the state-of-the-art performance via better architecture design, such as residual connections (Kim, Lee, and Lee 2016a,b; Liu, Tang, and Wu 2020; Liu et al. 2020; Wang et al. 2018), U-shape architecture (Cheng et al. 2019; Mao, Shen, and Yang 2016; Liu et al. 2018) and attention mechanisms (Hui et al. 2019; Zhang et al. 2018b; Zhao et al. 2020; Niu et al. 2020; Wu et al. 2020).

In addition to improving accuracy, some CNN-based works pursue a lightweight and economical structure. Specifically, IDN (Hui, Wang, and Gao 2018) stacks multiple information distillation blocks to extract more useful information. Each block divides feature maps into two parts. One part is reserved and the other is further enhanced by several convolutions. After that two kinds of features are combined for more abundant information. Based on information distillation block, IMDN (Hui et al. 2019) utilizes information multi-distillation block, which retains and process partial feature maps step-by-step, and aggregates them by contrast-aware channel attention mechanism. Furthermore, RFDN (Liu, Tang, and Wu 2020) applies intensive residual learning to distill more efficient feature representations.

While CNN-based methods have dominated this field for a long time, recent works introduce Transformer (Dosovitskiy et al. 2020) and make impressive progress. IPT (Chen et al. 2021) is one of the first methods use Transformer in image SR. However, IPT contains a massive amount of parameters and requires large-scale datasets for training, which limits its practice in real applications. Modified from Swin Transformer (Liu et al. 2021), SwinIR (Liang et al. 2021) limits the attention region in fixed-size windows and uses shift operation to exchange information through nearby windows. As such, it achieves a better trade-off between PSNR and the number of parameters than prevalent CNN-based models. Despite the success, SwinIR set the window size as $8 \times 8$, which is larger than a regular convolution (often

(Chen et al. 2021; Liang et al. 2021; Wang et al. 2021) to alleviate this issue. However, these methods lack direct interaction between distant pixels, which is critical for achieving good performance (Gu and Dong 2021). In (Gu and Dong 2021), a Local Attribution Map (LAM) is proposed to give a deep understanding of SR networks. As shown in Fig. 2b, the LAM map represents the importance of each pixel in the input LR image with respect to the SR of the patch marked with a red box (Fig. 2a). Based on the LAM map, we define a Area of Interest (AoI) map in Fig. 2c. The AoI map presents a hierarchical manner that the most important pixels are located in a fine area (yellow area in Fig. 2c) of a patch and some less important pixels are spread in a coarse area (gray area in Fig. 2c) of the whole image. Further, we use a blue box in Fig. 2c to represent the area with highest interest. In this paper, we use the terms coarse and fine to describe the density of important pixels and the granularity of operations interchangeably. According to this hierarchical partition, we believe that a well-designed SR network should (1) have the ability to focus on processing the finest detail in very local area (e.g., the blue area); (2) be capable of modeling long-range dependencies in a certain area (e.g., the yellow area) to utilize the surrounding context for better reconstruction; (3) have a mechanism to access important pixels in a coarse area (e.g., the gray area) of an input image.

Based on these observations, we propose a novel Hierarchical Pixel Integration (HPI) block that consists of three main parts: a Global Pixel Access (GPA) module, an Intra-Patch Self-Attention (IPSA) module, and a $3 \times 3$ convolution. The GPA module is responsible for pixel access in coarse area. Specifically, a similarity map between each pair of image patches is calculated and the most similar patch is selected to conduct cross-attention with current patch. In this way, important pixels in the coarse area can be integrated into current patch efficiently. After fusing pixels from coarse area, we apply the standard self-attention to model long-range dependencies in the fine area of a patch. Finally, a $3 \times 3$ convolution is adopted to refine local details in the finest area. Besides, we found in experiments that the vanilla patch division hinders the perceptual quality of recovered images. To tackle this issue, we propose to use a Cascaded Patch Division (CPD) that gradually enlarges the patch window in different blocks.

Figure 2: (a) SR result of RNAN (Zhang et al. 2019). (b) Local Attribution Map (LAM) (Gu and Dong 2021). (c) Area of Interest (AoI). The AoI is defined according to the coarse-to-fine hierarchy of important pixels in LAM.
is obtained by the CPD module and finally aggregated together via patch convolution (Conv). The input feature is divided into patches by respectively. We first get shallow feature \( x_0 \) by

\[
x_0 = f_{\text{Encoder}}(I_{LR}),
\]

where \( f_{\text{Encoder}} \) denotes the function of the encoder. Then deeper features are extracted by \( K \) sequential HPI blocks. The HPI block consists of five parts: Cascaded Patch Division (CPD), Global Pixel Access (GPA), Intra-Patch Self-Attention (IPSA), Patch Aggregation (PA), and \( 3 \times 3 \) Convolution (Conv). The input feature is divided into patches by the CPD module and finally aggregated together via patch aggregation. Formally, in the \( i_{th} \) block, the output feature \( x_i \) is obtained by

\[
x_i = x_{i-1} + f_{\text{Conv}}(f_{\text{PA}}(f_{\text{IPSA}}(f_{\text{GPA}}(f_{\text{CPD}}(x_{i-1}))))),
\]

where \( f(\cdot) \) denotes the function of each individual component. As with existing works, residual learning is used to assist the training process.

Finally, decoder with pixel-shuffle operations (Shi et al. 2016) is adopted to get the global residual information, which is added to \( I_{LR} \) for restoring the high-resolution output

\[
I_{SR} = I_{LR} + f_{\text{Decoder}}(x_K).
\]

### Cascaded Patch Division

In order to reduce the computational complexity for lightweight image SR, we split the input feature map into a collection of equal-sized patches and process each patch independently. Specifically, given an input \( X \) of size \( H \times W \times C \), we split it into a set of square patches, \( i.e., X = \{X_i \in \mathbb{R}^{P^2 \times C} \mid i = 1, \ldots, N\} \), where \( P \) is the patch size and \( N \) stands for the total number of patches. Each patch must satisfy

\[
u \% P = 0 \quad \text{or} \quad u = H - P \tag{4}
\]

\[
v \% P = 0 \quad \text{or} \quad v = W - P \tag{5}
\]

where \((u, v)\) denotes the coordinate of pixel in the top-left corner. All patches can be processed in parallel, after which the outputs are pasted to their original location in patch aggregation module. It is worthwhile to note that such cropping strategy is adaptive to arbitrary input size, which means no padding pixels are needed.

Though this vanilla division greatly reduces computational cost, it limits the receptive field of boundary pixels. Unlike pixels around the center of a patch, border pixels can not directly interact with neighboring pixels that are out of the patch (see Fig. 4a), which may deteriorate the visual quality of reproduced images. In practice, patch overlapping is useful but not effective enough. Other attempts like (Liang et al. 2021) use shift operations to re-divide patches, but there are still pixels that fail to reach their neighbors directly (see Fig. 4b). Inspired by the idea of progressive learning, we just assign different patch size \( P \) to different blocks in a cascaded manner. In other words, we enlarge \( P \) progressively in the network. As a result, the border pixels of a block could be in the center of later blocks. Therefore, there exist non-persistent boundaries (see Fig. 4c). It also comes with the added bonus that a smaller receptive filed in the shallow layers helps stabilize the training process, while a larger receptive field in deep layers enables smoother pixel integration. Experiments indicate that the Cascaded Patch Division (CPD) strategy is fairly simple yet effective.

### Intra-patch Self-attention

Before the introduction of our GPA module, we first describe the detail of Intra-Patch Self-Attention (IPSA) for a better understanding. IPSA follows the standard self-attention paradigm (Dosovitskiy et al. 2020), whereas there are two changes. Firstly, IPSA is performed at patch level instead of image level. Secondly, positional embedding is removed because of the introduce of the convolutional layer, which can learn positional relations implicitly and make the network
more concise and efficient. IPSA is responsible for modeling long-range dependencies in a patch so that the context information could be fully utilized.

More specifically, for a patch feature \( X \in \mathbb{R}^{p^2 \times C} \), the \textit{query}, \textit{key} and \textit{value} matrices \( Q \in \mathbb{R}^{p^2 \times d} \), \( K \in \mathbb{R}^{p^2 \times d} \), \( V \in \mathbb{R}^{p^2 \times C} \) are computed as

\[
Q = XW_Q, \quad K = XW_K, \quad V = XW_V,
\]

(6)

where \( W_Q, W_K \) and \( W_V \) are weight matrices that are shared across patches. By comparing the similarity between \( Q \) and \( K \), we obtain an attention map of size \( \mathbb{R}^{p^2 \times p^2} \) and multiply it with \( V \). Overall, the calculation of Self-Attention (SA) can be formulated as

\[
SA(X) = \text{softmax}(QK^T/\sqrt{d})V.
\]

(7)

Here \( \sqrt{d} \) is used to control the magnitude of \( QK^T \) before applying the softmax function.

Similar to the conventional Transformer layer (Dosovitskiy et al. 2020), the Feed Forward Network (FFN) is employed after SA module to further transform features. FFN contains two fully-connected layers, and one GELU non-linearity is applied after the first linear layer. Besides, layer normalization is added before SA module and FFN module, and residual shortcuts after both modules are added as well.

Global Pixel Access

In this part, we aim to integrate important pixels belonging to the coarse area of a LAM map (see Fig. 2). SR Networks with a wider range of effective receptive field have been proven to achieve better performance (Gu and Dong 2021). The problem is how to make the network be capable of modeling global connectivity while maintaining computational efficiency. Since the partition of patch is fixed for a layer, there is no direct connection across patches. A straightforward way is to mix the information of every patch pair exhaustively. However, it is unnecessary and inefficient given the fact that many patches are irrelevant and uninformative. Moreover, redundant interaction may introduce extra noise that hinders the model performance.

Based on these observations, we propose an innovative Global Pixel Access (GPA) module, in which each patch performs adaptive pixel integration with the most correlative counterpart (see Fig. 5). To be specific, firstly, all patches are spatially pooled into one-dimensional tokens. These tokens encode the characteristic of the patches, which are later used for similarity calculation and patch matching. This process can be expressed as

\[
\overline{X}_i = \arg \max_{j \neq i} L(X_i)^T L(X_j),
\]

(8)

where \( \overline{X}_i \) is the best-matching patch with \( X_i \), and \( L(\cdot) \) is the average pooling function along spatial dimension followed by flatten operation and layer normalization. Since the \textit{argmax} operation is non-differentiable, we replace it with Gumbel-Softmax operation (Jang, Gu, and Poole 2016) during training so as to make it possible to train end-to-end. After that, pixel information of \( \overline{X}_i \) are fused into \( X_i \) via Cross-Attention (CA)

\[
X_i = CA(X_i, \overline{X}_i).
\]

(9)

As illustrated in Fig. 5, CA works in a similar way to the standard self-attention (Dosovitskiy et al. 2020), but the \textit{key} and \textit{value} are calculated using \( \overline{X}_i \) in Equation (Eq. (6)). As a result, GPA can enable global pixel integration while introducing little computational overhead.

Local \( 3 \times 3 \) Convolution

The aforementioned GPA and IPSA can integrate information from a wide range of pixels. However, as indicates by the LAM map in Fig. 2b, the most import pixels usually locate in a local area of a patch. Therefore, we need a more local operation to process local detail in a fine-grained manner. We found that a simple \( 3 \times 3 \) convolution is efficient and effective for this task due to the local processing principle of convolution kernel.

Experiments

Datasets and Evaluation Metrics

The model is trained with a high-quality dataset DIV2K (Agustsson and Timofte 2017), which is widely used for image SR task. It includes 800 training images together with 100 validation images. Besides, we evaluate our model on five public SR benchmark datasets: Set5 (Bevilacqua et al. 2012), Set14 (Zeyde, Elad, and Protter 2010), B100 (Martin et al. 2001), Urban100 (Huang, Singh, and Ahuja 2015) and Manga109 (Matsui et al. 2017).

We use objective criteria, i.e., peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) to evaluate our model performance. As adopted in the previous work, the two metrics are both calculated on the Y channel after converting to YCbCr space. Besides, we report total number of parameters to compare the complexity of different models.
We compare the proposed HPINet with commonly used lightweight SR models for upsampling factor $\times 2$, $\times 3$, and $\times 4$, including FALSR (Chu et al. 2021), DRRN (Tai, Yang, and Liu 2017), A$^2$F (Wang et al. 2020), MAFFSRN (Muqeet et al. 2020), PAN (Zhao et al. 2020), IDN (Hui, Wang, and Gao 2018), IMDN (Hui et al. 2019), A$^2$F-L (Wang et al. 2020), and SwinIR-light (Liang et al. 2021).

Implementation Details
Motivated by (Zamir et al. 2021), we adopt progressive learning during training process. The cropped HR image size is initialized as 196$\times$196 and increases to 896$\times$896 epoch by epoch, and batch size is set as 6. Training images are augmented by random flipping and rotation. All models are trained using Adam algorithm with L1 loss. The learning rate is initialized as $3 \times 10^{-4}$ and halved per 200 epochs. For the proposed HPINet, the number of blocks is set as 8 and the corresponding patch size is set as $\{12, 16, 20, 24, 12, 16, 20, 24\}$. The whole process is implemented by Pytorch on NVIDIA Tesla V100 GPUs. More specific details will be explained in the respective subsection.

Comparisons with State-of-the-arts
We compare the proposed HPINet with existing lightweight SR models on benchmark datasets obtained by different algorithms. As shown in Table 1, our HPINet achieves the best performance for $\times 3$ and $\times 4$ SR. For $\times 2$ SR, we also achieve superior results than the state-of-the-art lightweight SR methods on Urban100, including IDN (Hui, Wang, and Gao 2018), CARN (Ahn, Kang, and Sohn 2018), IMDN (Hui et al. 2019), A$^2$F-L (Wang et al. 2020), and SwinIR-light (Liang et al. 2021).

Quantitative Comparison
Table 1 shows quantitative results in terms of PSNR on five benchmark datasets obtained by different algorithms. As shown in Table 1, our HPINet achieves the best performance for $\times 3$ and $\times 4$ SR. For $\times 2$ SR, we also achieve superior results than the state-of-the-art lightweight SR methods.

SwinIR-light model on Set14, B100 and Urban100 datasets with 0.1M fewer parameters. It is notable that our HPINet outperforms SwinIR-light by a maximum PSNR of 0.27dB, which is a significant improvement for image SR.

Qualitative Comparison
We further show visual examples of different methods under scaling factor $\times 4$. As shown in Fig. 6, our HPINet can recover more details than IDN, CARN, IMDN, A$^2$F-L and SwinIR-light, which indicates the superiority of our method.

Ablation Analysis
In this section, we conduct ablation experiments to study the effect of Cascaded Patch Division (CPD) and Global Pixel Access (GPA). Evaluations are performed on Set5 (Bevilac-
Table 2: Ablation study on the Cascaded Patch Division (CPD) and Global Pixel Access (GPA). Metrics (PSNR↑/SSIM↑/LPIPS↓) are calculated on benchmark datasets with a scale factor of 4, where “↑” indicates higher is better and “↓” means lower is better.

<table>
<thead>
<tr>
<th>Model</th>
<th>CPD</th>
<th>GPA</th>
<th>Param</th>
<th>MACs</th>
<th>Set5</th>
<th>Set14</th>
<th>B100</th>
<th>Urban100</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>PSNR↑/SSIM↑/LPIPS↓</td>
<td>PSNR↑/SSIM↑/LPIPS↓</td>
<td>PSNR↑/SSIM↑/LPIPS↓</td>
<td>PSNR↑/SSIM↑/LPIPS↓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>①</td>
<td>×</td>
<td>×</td>
<td>895K</td>
<td>121.7G</td>
<td>32.46/0.8974/0.1760</td>
<td>28.83/0.7859/0.2844</td>
<td>27.70/0.7400/0.3444</td>
<td>26.57/0.7995/0.2125</td>
</tr>
<tr>
<td>②</td>
<td>✓</td>
<td>×</td>
<td>895K</td>
<td>123.9G</td>
<td>32.50/0.8979/0.1735</td>
<td>28.83/0.7869/0.2817</td>
<td>27.71/0.7416/0.3421</td>
<td>26.63/0.8018/0.2094</td>
</tr>
<tr>
<td>③</td>
<td>×</td>
<td>✓</td>
<td>896K</td>
<td>121.8G</td>
<td>32.50/0.8977/0.1757</td>
<td>28.86/0.7865/0.2868</td>
<td>27.72/0.7408/0.3476</td>
<td>26.63/0.8006/0.2163</td>
</tr>
<tr>
<td>④</td>
<td>✓</td>
<td>✓</td>
<td>896K</td>
<td>124.0G</td>
<td>32.60/0.8986/0.1729</td>
<td>28.87/0.7874/0.2826</td>
<td>27.73/0.7419/0.3433</td>
<td>26.71/0.8043/0.2100</td>
</tr>
</tbody>
</table>

Figure 7: LAM (Gu and Dong 2021) comparison between the full HPINet (w/ GPA) and the variant without GPA (w/o GPA) for × 4 SR. The first row shows input along with the predicted result, and the second row shows the effective receptive field. Zoom in for better view.

Figure 8: Visualization of patch pairs that best match. Patches with different size come from different blocks in the network.

(a) w/ GPA  (b) w/o GPA

Effectiveness of CPD  To show the effect of CPD, we instantiate model ③ where all blocks share the same patch size for comparison. The patch size is fixed as 18 to maintain similar Multi-Adds. As shown in Table 2, model ④ improves the performance of model ③ on all metrics. The same phenomena can also be observed by comparing model ② and model ③, where the LPIPS gets substantially improvement by adding CPD. It is notable that the model with CPD turns out to achieve better SSIM and LPIPS, even when its PSNR results are indistinguishable. These results validate that CPD can provide not only higher image similarity but also better perceptual quality.

Effectiveness of GPA  A core feature of our HPINet is its ability to access global pixel effectively. To highlight the contribution of global modeling, we drop GPA in model ② for comparison, meanwhile expanding IPSA modules to keep similar parameter budget. Recall that IPSA only integrates pixels within a patch. Comparing model ③ and model ④, we can observe that with the help of GPA, the PSNR and SSIM get consistent improvements on all five datasets. Specifically, the PSNR increases by a maximum of 0.1dB, which is a notable boost in lightweight image SR. Interestingly, the LPIPS on several benchmark datasets gets worse after adding GPA to the baseline model, and we guess it is caused by the introduction of noisy pixels. This problem can be greatly reduced by using the proposed CPD module, which can be observed by comparing model ③ with model ④.

To better understand the main reason of the improvement brought by GPA, we utilize LAM (Gu and Dong 2021) to visualize the effective receptive field of an input patch. As shown in Fig. 7, the patch benefits from a global range of useful pixels by using GPA. In Fig. 8, we further present examples of patch pairs that matched and selected by GPA in different blocks. It can be found that each patch pair share some visual similarities, which could facilitate restoring more details. All results indicate the effectiveness of the proposed GPA in improving PSNR and SSIM performances.

Effectiveness of IPSA  The IPSA module can model long-range dependencies in a patch. To prove the effectiveness of this module, we remove IPSA from all the attention blocks and replace it with depth-wise convolutions to maintain a similar number of parameters. As shown in the first row of
Table 3: Parameter, Running Time and PSNR comparison for scale factor ×4.

<table>
<thead>
<tr>
<th>Model</th>
<th>Param(M)</th>
<th>Set5</th>
<th>Set14</th>
<th>B100</th>
<th>Urban100</th>
<th>Manga109</th>
</tr>
</thead>
<tbody>
<tr>
<td>RCAN</td>
<td>15.6</td>
<td>32.63/56</td>
<td>28.87/65</td>
<td>27.77/61</td>
<td>26.82/119</td>
<td>31.22/157</td>
</tr>
<tr>
<td>SAN</td>
<td>15.9</td>
<td>32.64/68</td>
<td>28.92/127</td>
<td>27.78/58</td>
<td>26.79/997</td>
<td>31.18/1771</td>
</tr>
<tr>
<td>SwinIR</td>
<td>11.9</td>
<td>32.72/65</td>
<td>28.94/117</td>
<td>27.83/75</td>
<td>27.07/401</td>
<td>31.67/521</td>
</tr>
<tr>
<td>SwinIR-light</td>
<td>0.90</td>
<td>32.44/47</td>
<td>28.77/59</td>
<td>27.69/48</td>
<td>26.47/158</td>
<td>30.92/198</td>
</tr>
<tr>
<td>HPINet-L</td>
<td>1.44</td>
<td>32.72/39</td>
<td>28.97/51</td>
<td>27.79/40</td>
<td>26.95/144</td>
<td>31.47/220</td>
</tr>
<tr>
<td>HPINet-M</td>
<td>0.90</td>
<td>32.60/38</td>
<td>28.87/49</td>
<td>27.73/39</td>
<td>26.71/137</td>
<td>31.19/194</td>
</tr>
<tr>
<td>HPINet-S</td>
<td>0.46</td>
<td>32.47/34</td>
<td>28.80/48</td>
<td>27.69/36</td>
<td>26.59/131</td>
<td>30.92/160</td>
</tr>
</tbody>
</table>

Table 4: Ablation study on IPSA and 3 × 3 Conv. The PSNR results on five benchmark datasets are included.

<table>
<thead>
<tr>
<th>Model</th>
<th>Set5</th>
<th>Set14</th>
<th>B100</th>
<th>Urban100</th>
<th>Manga109</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o IPSA</td>
<td>32.26</td>
<td>28.66</td>
<td>27.60</td>
<td>26.30</td>
<td>30.66</td>
</tr>
<tr>
<td>w/o 3 × 3 conv</td>
<td>32.09</td>
<td>28.63</td>
<td>27.57</td>
<td>26.17</td>
<td>30.55</td>
</tr>
<tr>
<td>HPINet (full)</td>
<td>32.60</td>
<td>28.87</td>
<td>27.73</td>
<td>26.71</td>
<td>31.19</td>
</tr>
</tbody>
</table>

Effectiveness of 3 × 3 Conv The HPI attention block is designed in a hierarchical manner that the tail 3 × 3 convolution is responsible for processing the finest local details. As shown in the second row of Table 4, the model “w/o 3 × 3 Conv” achieves much lower PSNR values than the full model, which meets the expectation. It indicates convolution plays a basic role even in Transformer-like models, which is in line with conclusions from other vision tasks (Wu et al. 2021; Guo et al. 2022; Yuan et al. 2021). It is worth emphasizing that convolution is indispensable but not competitive enough. For example, with the equipment of IPSA and GPA, our proposed HPINet surpasses the pure convolution-based model RCAN (Zhang et al. 2018b) in all datasets with 10 × fewer parameters (See Table 3). Thus it is hard to replace IPSA and GPA with convolution while maintaining similar performance. Together with the aforementioned results, the observation proves effectiveness of our hierarchical design.

Model Size and Running Time Analyses

To demonstrate the effectiveness and efficiency of HPINet, we design three variants with different model size (S/M/L) and evaluate their PSNR results and speed on five datasets. For simplicity, All variants only differ in number of channels.

Model Size The curve of PSNR v.s. model size is depicted in Fig. 1 and the detailed complexity of the three models are included in Table 3. We compare our HPINet with other lightweight SR models of various sizes, including MAFFSRN (Muq et al. 2020), RFDN (Liu, Tang, and Wu 2020), LAPAR-A (Li et al. 2020), IMDN (Hui et al. 2019), LatticeNet (Luo et al. 2020), SwinIR-light (Liang et al. 2021), A$^2$-F-L (Wang et al. 2020) and A-cubeNet (Hang et al. 2020). Our HPINet-S/M/L achieves much higher PSNR than all other lightweight models at each size. Specially, HPINet-L outperforms RCAN (15.6M) and SAN (15.9M) with less than 1.5M parameters.

Running Time To reduce the accidental error, we run each model for 10 times on one GPU and calculate the average time as the final running time. We also compare them with other advanced models, including RCAN (Zhang et al. 2018b), SAN (Dai et al. 2019) and SwinIR (Liang et al. 2021). RCAN (Zhang et al. 2018b) is a classic CNN-based model and SAN (Dai et al. 2019) is a often-cited model equipped with non-local modules. SwinIR (Liang et al. 2021) is a state-of-the-art Transformer-based model. According to Table 3, several observations can be summarized as follows: (1) With the same (i.e., HPINet-M) or fewer parameters (i.e., HPINet-S), our model runs faster than SwinIR-light while maintaining higher PSNR values. (2) When using a slightly larger model, i.e., HPINet-L, we can even achieve superior or comparable PSNR values with very large models including RCAN, SAN and SwinIR. (3) Our HPINet runs faster than all other models on Set5, Set14 and B100 datasets. As the image resolution increases on Urban100 and Manga109 datasets, HPINet is slightly slower than RCAN but much faster than SAN and SwinIR. Overall, the proposed HPINet has a better trade-off between model complexity and PSNR.

Conclusion

In this paper, we proposed a lightweight single-image super-resolution network called HPINet which sequentially stacks a series of Hierarchical Pixel Integration (HPI) blocks. The HPI block is designed according to the hierarchical interpretation of a LAM map. Specifically, each block consists of three main components: a Global Pixel Access (GPA) module, an Intra-Patch Self-Attention (IPSA) module and a 3 × 3 convolutional layer. They are responsible for processing input images from coarse to fine. Besides, a Cascaded Patch Division (CPD) strategy is also proposed for better perceptual quality. Benefiting from these components, HPINet can effectively capture the global, long-range and local relations in an efficient manner. Experimental results demonstrate the superior performance of HPINet over previous state-of-the-art SR models on benchmark datasets. In the future, we will investigate the potential of HPINet in other low-level tasks.
Acknowledgments

This work is supported by National Natural Science Foundation of China (No.62076119, No.61921006, No.62072232), Program for Innovative Talents and Entrepreneur in Jiangsu Province, and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References


